Egy kereslet-kínálat elvű elemző működése és a koordináció kezelésének módszere

Sass Bálint

MTA-PPKE Magyar Nyelvtechnológiai Kutatócsoport, PPKE ITK
sass.balint@itk.ppke.hu

Korábban általánosságban bemutattuk pszicholingvisztikai indíttatású, performanciaalapú, szigorúan balról jobbra haladó magyar nyelvi elemzőnk elvi megalapozását [1]. Az igei vonzatkeretek és a tematikus szerepek kezeléséről jelen kötetben olvashatunk [2]. Most részletesebben tárgyaljuk a szűken vett központi elemző (és működtető) komponenst, illetve ennek azt a részét, ami konkrétan a balról jobbra haladást/elemzést végzi: egyesével végiglépkedve a szavakon egyfajta függőségi szintaktikai elemzésnek veti alá a szöveget.

A bemenet tokenek sorozata, ezeket veszi sorra az elemző. A balról jobbra PoS-taggernek és egyéb párhuzamosan futó komponenseknek (erőforrásszálaknak [1]) köszönhetően a szóalakon és a szótőn túl számos egyéb információ is rendelkezésre áll minden tokenhez. A tokeneket (és a belőlük képzett nagyobb egységeket, frázisokat) egy *stack*en tároljuk. Az elemző jelenleg kézzel írt *szabályok* alapján dolgozik. Egy szabály egy feltételt és egy eljárást határoz meg, ha az adott tokenre teljesül (egy vagy több) feltétel, akkor lefut a hozzá rendelt eljárás. Az eljárások tipikusan az alábbi három lépésből állnak: (1) az adott token bekerül a stackre; (2) a stackelem lezárul, ha frázis végén vagyunk; (3) az adott tokennek megfelelően valamilyen strukturális *szál* indul/lezárul, vagy valamilyen (jelenleg egyfajta: alárendeltséget kifejező függőségi) *él* keletkezik két stackelem között. Ezen élek segítségével tudunk fákat (gráfokat) építeni a stacken lévő elemek között. A strukturális szálak (röviden: szálak) két típusát különböztetjük meg: a *felkínálás*, illetve az *igény* jellegű szálat [1]. Az elváló igekötő például igét igénylő szálat indít, az igekötőmentes ige egy felkínáló szálat (az esetleges igekötő részére), a szálak típusa természetesen független az ige és igekötő sorrendjétől. Az épp futó szálakat egy halmazban tartjuk nyilván, ehhez a halmazhoz fordul az elemző minden egyes token feldolgozásakor. A (tag)mondat végén az elemző értékelést készít, összesíti a (tag)mondatban lévő felsőszintű elemeket. E reprezentáció segítségével fogjuk tudni megvalósítani a kitűzött célt, hogy a szöveg alapján válaszolni tudjunk az olyan kérdésekre, hogy ki, mit csinált, hol és mikor.

A párjukat kereső felkínálás-igény szálakat tartalmazó architektúránk sokban hasonlít a Link Grammar [3] felépítéséhez, de sokban el is tér tőle (performanciaközpontúság vs. kompetencia-központúság; láncolás vs. fejhez kötés; kategoriális vs. lexikalizált stb.). Már ez a klasszikus cikk nehézségként említi a koordinációkat: bizonyos koordinációk a Link Grammar eszközeivel – a linkek kereszteződése miatt – közvetlenül nem kezelhetők. Egy friss magyar tanulmány [4], mely az adatvezérelt függőségi elemzés hibaelemzésével foglalkozik, számos más probléma mellett – talán legnehezebbként – szintén a koordináció kérdését emeli ki. Más problémákkal ellentétben a koordináció kezelésére nem is ad javaslatot a

cikk, hanem a szemantika területére utalja, azzal érvelve, hogy a koordinációk gyakran csak kontextuális vagy szemantikai háttértudás segítségével értelmezhetők helyesen.

A továbbiakban arról lesz szó, hogy hogyan kezeljük a koordinációt a vázolt architektúrában. Alapelv, hogy a felsorolás valamilyen értelemben azonos típusú elemekből áll. Az elemzőben a koordinációt (felsorolást) egy speciális fajtájú felkínáló szál segítségével kezeljük. Ez a szálfajta tartalmaz egy külön adattagot (*pattern*), mely a felsorolás kezdete óta feldolgozott felső szintű egységekről szóló információt (PoS-tag stb.) tárolja annak érdekében, hogy ellenőrizni tudjuk az említett alapelv teljesülését. A konjunktív elemeknél (*és*, *vagy*, vessző) elindítunk egy ilyen felsorolás-szálat, úgy, hogy természetesen a konjunktív elemet megelőző egységről szóló információt is hozzávesszük. A szál külső (pl. névutó, *és* utáni vessző megjelenése, mondat vége) vagy belső esemény (pl. *A és A* alakú *pattern*) hatására zárul le. Lezáráskor kiértékelődik a *pattern* adattag: nem egyszerű azonosságot vizsgálunk, például egyes szám harmadik személyű ige és alanyesetű melléknév megengedett ugyanazon felsoroláson belül. A beazonosított felsorolásokból egy új komplex elemet képezünk a stacken, a komplex elem feje a koordinált elemek fejeinek összessége lesz. Az új komplex elem egy egységként funkcionál aztán tovább, akár egy újabb koordinációs szerkezet egyik tagjaként.

Alább néhány példmondat elemzésének eredményével illusztráljuk a fentieket.

- *„A kutya megkergette és megharapta Marit ."* ([3] problematikusként említett példájának fordítása.) A felsorolás lezárulását az *A és A* mintázat váltja ki, a mondat alanyt, tárgyat és állítmányt (*megkergette+megharapta*) tartalmaz.
- *„Józsi és Pisti mellett nem fut el ."* A felsorolás lezárását itt a névutó váltja ki, a felsorolás feje/lemmája a *Józsi+Pisti* lesz, és utána a felsorolás egységhez kapcsolódik hozzá a névutó.
- *„Aláírják a finanszírozási szerződést a Budapesti Közlekedési Központ igazgatósága és a Fővárosi Közgyűlés jóváhagyásával ."* Az egy egységként azonosított terjedelmes koordinációhoz járul hozzá a birtok: a mondatban állítmány, tárgy és *-vAl* ragos bővítmény (valamint kikövetkeztetett alany) van.
- *„Romulus és Remus, Róma későbbi két városalapítója egy fügefa árnyékában szopta a farkasanya tejét ."* A *Romulus és Remus* koordináció lezárul a követő vessző hatására, majd első eleme lesz ugyanezen vessző által indított másik koordinációnak. Végül ez utóbbi értelmezőként (*Róma későbbi két városalapítója*) elemződik.

Hivatkozások

1. Prószéky, G., Indig, B., Miháltz, M., Sass, B.: Egy pszicholingvisztikai indíttatású számítógépes nyelvfeldolgozási modell felé. In Tanács, A., Varga, V., Vincze, V., eds.: X. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY2014), SZTE, Szeged (2014) 79–87
2. Miháltz, M., Indig, B., Prószéky, G.: Igei vonzatkeretek és tematikus szerepek felismerése nyelvi erőforrások összekapcsolásával egy kereslet-kínálat elvű elemzőben. In: XI. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY2015). (2015) 298–302

3. Sleator, D., Temperley, D.: Parsing English with a link grammar. In: Proceedings of the Third International Workshop on Parsing Technologies. (1993)
4. Farkas, R., Vincze, V., Schmid, H.: Dependency parsing of Hungarian: Baseline results and challenges. In: Proceedings of the 13th Conference of the EACL, Avignon, France (2012) 55–65